SensePresence: Infrastructure-less Occupancy Detection for Opportunistic Sensing Applications

16th IEEE International Conference on Mobile Data Management

Md Abdullah Al Hafiz Khan   H M Sajjad Hossain   Nirmalya Roy

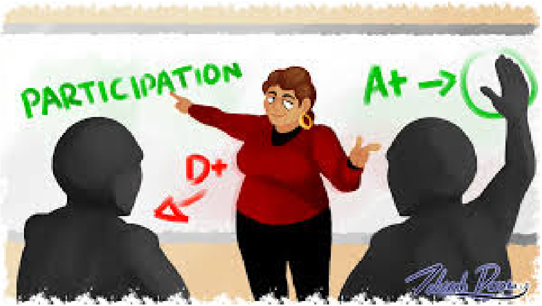
University of Maryland Baltimore County
Mobile, Pervasive & Sensor Computing Lab
Universities of Maryland Baltimore County

June 15, 2015

Outline

1. Motivation
   - Class Participation
   - Social Gathering
   - Group member's participation
2. Solutions
3. Ubiquitous Voice Sensing
   - Ubiquitous Sensing
   - What we have already?
   - Voice Centric Sensing
4. Goals and Challenges
5. Overview of SensePresence
6. Methodology
   - Speaker Counting Algorithm
   - Locomotive Counting
7. Experimental Setup and Results
8. Discussion & Future Work

Which class is interactive?



- Helps to solve problems and theories.
- Helps gain knowledge.
- Total interactive participants.

How many people are there?

- Is the party enjoyable?

Group member's participation

- How many people participate in the meeting?
- Does all the member participate?

Solutions

- People Count!
  - Which class is Interactive?
    - Check how many students ask questions?
  - Where is the party?
    - Find the place where most people speaks.
  - Is the meeting effective?
    - How many members participate?
- Microphone + Accelerometer Sensors

Ubiquitous Sensing

Smart Watch

Brace

Necklace

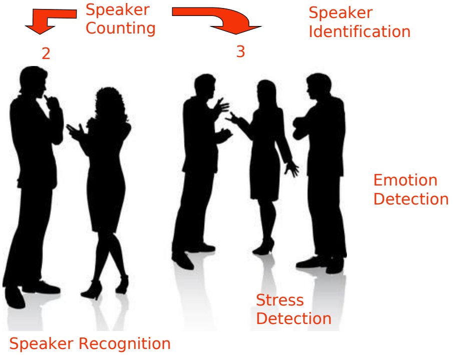
Phone

- What are the sensors available today?
- Which smart devices belongs to all people?

What we have already?

- Accelerometer
- Microphone
- Gyroscope
- etc.

Voice Centric Sensing

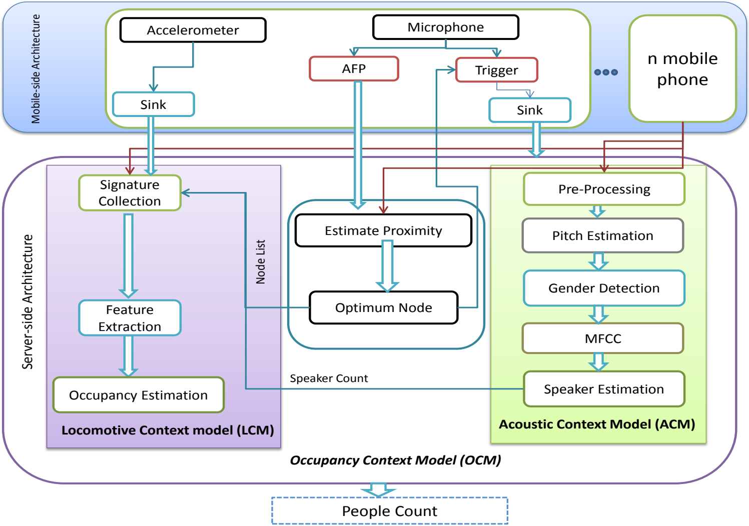


- What are the different types of application using voice centric sensing?
- “Blind Speaker clustering”, Iyer, IEEE, ISPACS (2006)
- “Crowd++: Unsupervised speaker count with smartphones”, Chenren Xu, UbiComp (2013): Static segmentation, controlled scenario where all speakers are active

Goals and Challenges

Challenges	Solution
No prior knowledge of speakers	Best Feature Extraction
Background noise	Filter
Some people might remain silent	Other Sensor (Accelerometer)
Speech overlap	Overlap Detection
Privacy concern	Use encryption (steganographic,steg

SensePresence Architecture

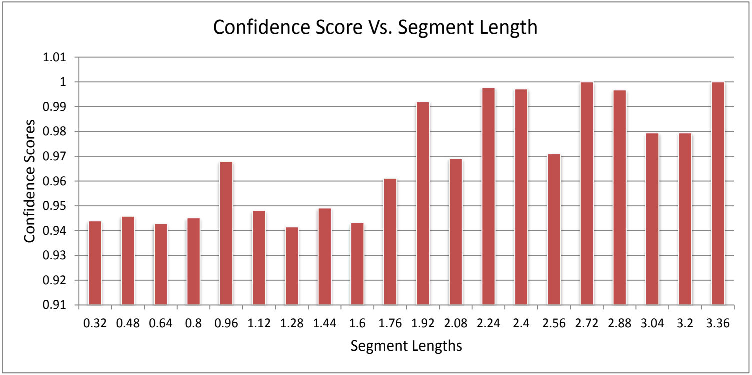


Methodology

- Acoustic methodology
  - Create segment from raw audio
  - Find Male and Female Segments
  - Audio Processing
- Locomotive methodology
  - Select sensors data based on speaker count and node list
  - Calculate Magnitude
  - Detect abrupt changes on the signal

Case 1: when people are conversing

Dynamic Segmentation

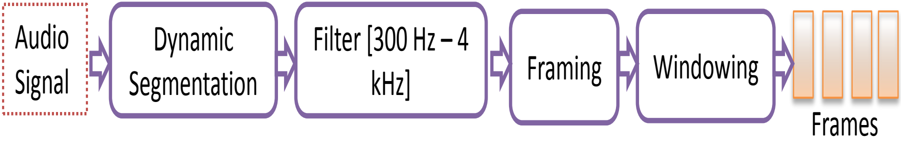


- What is the minimum or maximum segment length?
- Consider higher confidence score
- Which segment to choose when multiple segments have same confidence (i.e. 2.72 vs. 3.36 seconds)

Gender Detection

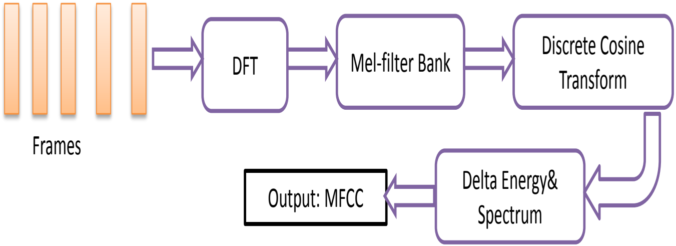
- Calculate Pitch
- Human voice ranges from 50Hz to 450Hz
- Male pitch falls between 100Hz to 146Hz
- Female pitch falls between 188Hz to 221Hz.
- Make Male and Female Segment sets.

Audio Processing



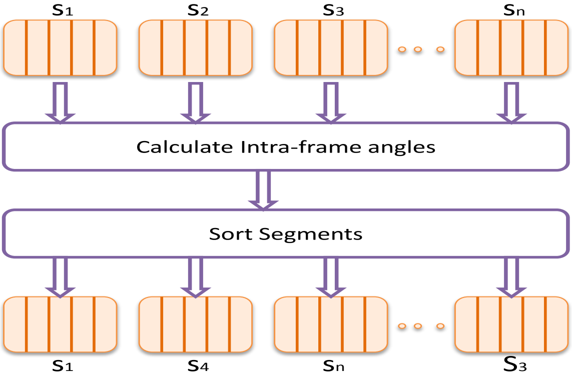
- Hamming window (50% overlapped)
- Frame length 32 ms
- Band pass filter (300Hz - 4000Hz)

Mel-frequency cepstral coefficients



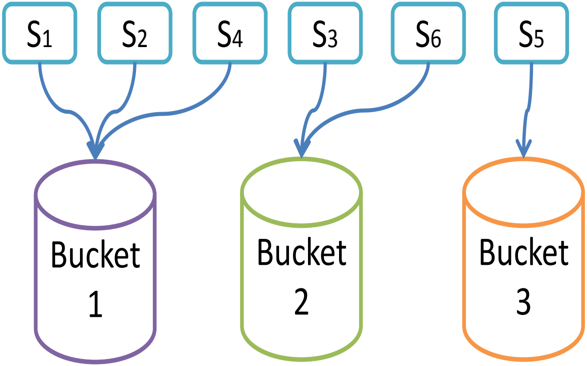
- Take Fourier transform
- Apply triangular mel-filter bank to map the power of the spectrum and take log
- Apply Discrete cosine transform
- Amplitude of the spectrum is the MFCC

Segment Sets Sorting



- Calculate intra-frame cosine angles
- Take average intra-frame angles
- Sort segment based on avg. angle

Grouping of Human Speakers based on Proximity

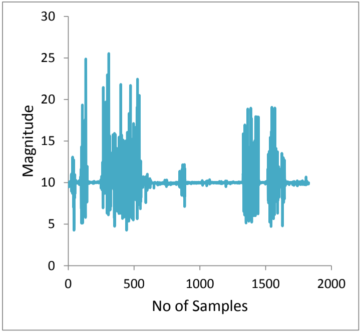


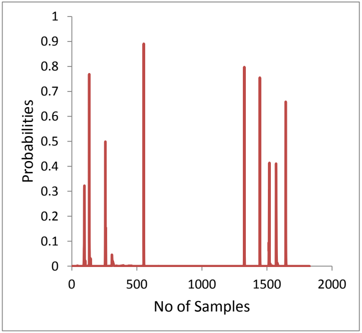
- Calculate inter-frames cosine distance
- For similar person distance is less than equal 15 degree

Case 2: People are not Conversing

- Change point to capture the locomotive movements
- Use change points to find stray movements
- Baysian changepoint detection algorithm
  - Calculate a-priori probability of two succesive change points at distance d (run length)
  - Gaussian based log-likelihood model to compute log-likelihood of the data sequence [s,d] where no change point has been detected.
  - Calculate log-likelihood for the entire signal S[t,n], log-likelihood of data sequence $S_s[t, s]$ where no change point occurs, $\pi[i, t]$ log-likelihood where change point occurs
  - summing up log-likelihoods for that sequence at time t
  - set threshold δ_{th}
  - Count number of change points to assign movement score

Change Point Detection Result





- Change point with probability values
- Count the number of changepoint as movement score
- Set threshold probability to eliminate few changepoint

Experimental Setup and Results

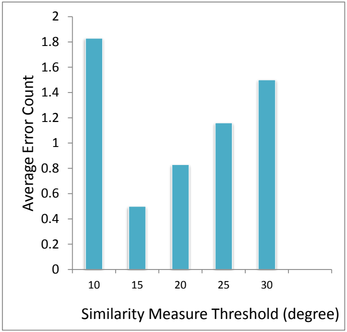
Data Collection:

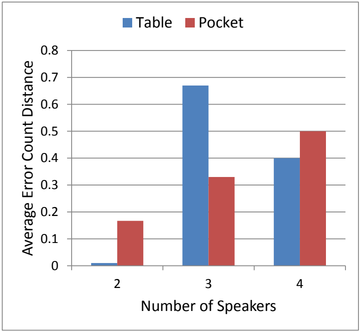
- Natural conversation data collected and make it properly anonymized
- lab meeting, general discussion in lobby/corridor
- Data collection was 1-10 persons (with 5 males and 5 females) with age group of 18-50 years
- Audio sampling rate 16kHz at 16 bit PCM
- Locomotive sampling rate 5kHz

Evaluation Metric:

- We use the average error count as the normalized predicted occupancy metric
- **Error Count:** $\frac{|EC-AC|}{N}$
- where EC, AC, N denote the estimated people count, actual people count and number of samples respectively
- We use absolute value in order to avoid any positive or negative contribution

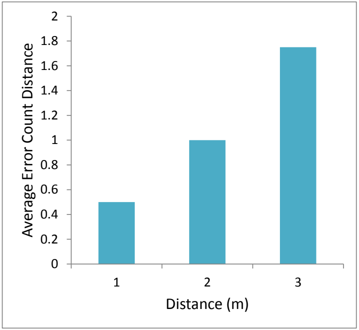
Occupancy Counting Results

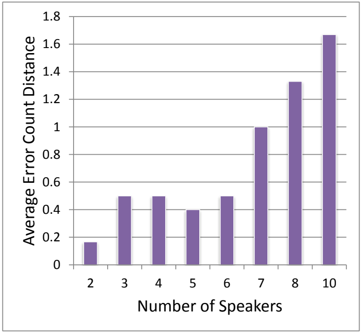




- Left figure depicts the effect of cosine distant similarity measures
- Similarity measure threshold is 15 degree
- Right figure reports the average error count distance 0.5 with respect to different phone positions

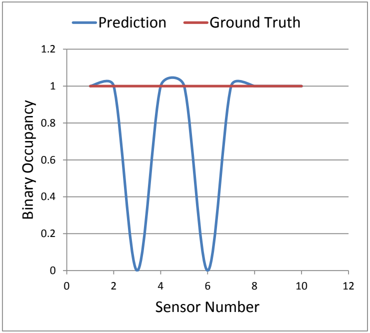
Occupancy Counting Results

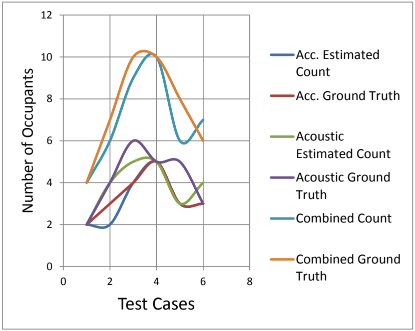




- Left figure depicts that error count increases as leader's distance from other occupants increases
- Right figure presents speaker counting performance (both overlapped and non-overlapped conversation)

Occupancy Counting Results





- Left figure shows binary occupancy counting
- Right figure presents locomotive augmented acoustic occupancy counting
- Example, 6 people converse and 4 remains silent. Acoustic sensing estimates 5 and locomotive sensing estimates 4. So total occupancy 9 out of 10 people

Comparison with existing methodology

Number of Speakers	Crowd++ (Error Count)	Sense Presence (Error Count)
2	0.5	0.167
4	2.33	0.5
6	2.5	0.83
Average	1.78	0.5

- Average error count distances for Crowd++ and SensePresence
- SensePresence accuracy increase more than 3 fold of Crowd++

Discussion & Future Work

- Innovative system to infer number of people in a location.
- Unsupervised speaker count
- Posit changepoint detection algorithm to detect binary occupancy
- Context aware client-server based architecture
- Use smartphone's microphone and accelerometer to count people
- Average error count 0.76
- In future, we will Explore energy consumption
- Will try to add modality by adding location information
- Privacy issues can be resolved

Thank You

Md Abdullah Al Hafiz Khan
Mobile Pervasive & Sensor Computing Lab
University of Maryland Baltimore County
email: mdkhan1@umbc.edu
Website: http://userpages.umbc.edu/ mdkhan1/